

(c) Осипов Егор (Б03-005)

Square equation solver

Bug report and messages: osipov.ea@phystech.edu

Functions and their usage:

- `void instructions()` puts instructions from file `<./instructions.txt>` in console.
- `void elementary_error()` puts in console a warning message about wrong input.
- `char is_equal(double a, double b)` returns 1 if `a` is equal to `b` and 0 otherwise
- `char is_bigger(double a, double b)` returns 1 if `a` is bigger then `b` and 0 otherwise
- `char is_less(double a, double b)` returns 1 if `a` is less then `b` and 0 otherwise
- `int get_equation_result(double a, double b, double c, double* x1, double* x2)` writes in `x1` and `x2` solutions of equation with coefficients `a`, `b` and `c`. Returns count of roots and -1 if count of roots is infinite
- `double calculate_the_discriminant(double a, double b, double c)` Calculates a discriminant of equation with coefficients `a`, `b` and `c`. Returns -1 if discriminant less then 0
- `double calculate_linear_equation(double b, double c)` calculates `(-c / b)` if `b` is not zero, otherwise continue with aborting.
- `void unit_tests()` A set of tests.... nothing to say there
- `void check_unit_test(double x1, double x2, int nroots, const double tx1, const double tx2, const int tnroots, char* p, int test_number)` checks if test number `test_number` passed. `x1`, `x2`, `nroots` is the first and the second solutions and number of roots that were received from `get_equation_result()`. `tx1`, `tx2`, `tnroots` - correct answer. If unit test is not passed, `p` becomes 0 and `unit_tests()` won't print "Tests passed without problems".
- `void print_result(double x1, double x2, int number_of_roots)` The function that prints an answer depending on number of roots.

Main variables

- `a`, `b`, `c` - coefficients of equation
- `number_of_roots` - count of solutions
- `command` - user's input. Depending on it program running `unit_tests()` or `instructions()` or nothing
- `x1`, `x2` - variables in which solutions are stored
- `D` - variable for discriminant
- `passed` - variable in `unit_test()` which shows if at least one test is not passed

Решатель квадратных уравнений

О багах или просто о своем довольстве программой можно сообщать мне на почту: osipov.ea@phystech.edu

Функции и их назначение

- `void instructions()` Выводит инструкции из файла `<./instructions.txt>` в консоль
- `void elementary_error()` Выводит в консоль предупреждение о неправильном формате ввода
- `char is_equal(double a, double b)` возвращает 1, если `a` равно `b`, и 0 в другом случае

- `char is_bigger(double a, double b)` Возвращает 1, если `a` больше `b` и 0 в другом случае
- `char is_less(double a, double b)` Возвращает 1 если `a` меньше `b` и 0 в другом случае
- `int get_equation_result(double a, double b, double c, double* x1, double* x2)` Записывает в `x1` и `x2` решения уравнения с коэффициентами `a`, `b` и `c`. Возвращает количество корней и -1, если их бесконечно много
- `double calculate_the_discriminant(double a, double b, double c)` Вычисляет дискриминант уравнения с коэффициентами `a`, `b` and `c`. Возвращает -1, если дискриминант меньше 0
- `double calculate_linear_equation(double b, double c)` Вычисляет $(-c / b)$ если `b` не равно 0, в противном случае закончивается `abort()`ом
- `void unit_tests()` Просто набор тестов
- `void check_unit_test(double x1, double x2, int nroots, const double tx1, const double tx2, const int tnroots, char* p, int test_number)` Проверяет, пройден ли тест с номером `test_number`. `x1`, `x2`, `nroots` - первое и второе решения, а также количество корней уравнения, полученные из `get_equation_result()`. `tx1`, `tx2`, `tnroots` - Правильный ответ. Если тест не пройден, `p` становится 0 и `unit_tests()` не напишет "Tests passed without problems".
- `void print_result(double x1, double x2, int number_of_roots)` Функция, которая выводит ответ в зависимости от количества корней

Основные переменные

- `a`, `b`, `c` - коэффициенты уравнения
- `number_of_roots` - количество корней уравнения
- `command` - входные данные. в зависимости от того, что хранится в `command` программа выполняет функции `unit_tests()` или `instructions()` или ничего
- `x1`, `x2` - переменные, в которых хранятся решения уравнения
- `D` - переменная, в которую записывается дискриминант
- `passed` - переменная в функции `unit_test()`, которая обнуляется. если хотя бы один тест не пройден